JOURNAL OF COMPLEXITY 1, 11–44 (1985)

A Survey of Information-Based Complexity*,†

H. WOŹNIAKOWSKI

Department of Computer Science, Columbia University, New York, New York 10027; and Institute of Informatics, University of Warsaw, 00-901 Warsaw, Poland

We survey some recent results in information-based complexity. We focus on the worst case setting and also indicate some average case results.

1. INTRODUCTION

The purpose of this paper is to give the reader some flavor of information-based complexity, to survey some recent results, and to indicate future research directions.

Information-based complexity is the study of the intrinsic difficulty of solving problems for which the information is *partial, contaminated,* and *priced.*

Problems for which only such information is available may be found in diverse areas including: mathematical economics, image understanding, applied mathematics, decision theory, numerical analysis, and many branches of science and engineering. If the information is partial and/or contaminated the problem can be only approximately solved.

* Presented at the Symposium on Complexity of Approximately Solved Problems, April 17, 1985.

† This research was supported in part by the National Science Foundation under Grant DCR-82-14322.

0885-064X/85 $3.00

An expository account of information-based complexity may be found in Traub and Woźniakowski (1984a). The reader is also referred to the papers of Traub (1985a,b), where complexity of approximately solved problems and the role of information are discussed from a general point of view.

We now illustrate information-based complexity by three examples. In each example we define problem, information, and model of computation.

EXAMPLE 1.1 (Continuous Binary Search). This is a generalization of the twenty questions game. The reader is asked to think of a real number x from the interval $F = (0, 1)$. Our aim is to find the number x to within a prescribed accuracy ϵ, $\epsilon > 0$, by asking questions whose answers are either true or false. We assume that any such question is allowed and that each question has a fixed cost of c. Moreover, we can perform arithmetic operations and comparisons of real numbers. We assume that such operations cost unity and are performed exactly; i.e., we use the real number model of computation.

We wish to find the number x to within ϵ with minimal cost. The continuous binary search problem is formalized as follows:

Problem. For each x in $F = (0, 1)$, find x_ϵ such that $|x - x_\epsilon| \leq \epsilon$, $\epsilon \in (0, 1)$.

Information. (i) We know a priori that $x \in F = (0, 1)$.

(ii) We can ask questions about subset membership; i.e., for an arbitrary subset T of F we can compute

$$Q(x, T) = \begin{cases} 1 & \text{if } x \in T, \\ 0 & \text{if } x \notin T. \end{cases}$$

Model of computation. (i) Each question costs c, $c > 0$.

(ii) We can perform arithmetic operations and comparisons in the real number model. Each costs unity.

What is the ϵ-complexity of the continuous binary search problem; i.e., what is the minimal cost of finding x_ϵ? Which questions should be asked and how should the answers be combined in order to find x_ϵ with minimal cost?

Let $\text{comp}(\epsilon)$ denote the ϵ-complexity. It is easy to check that we have to ask at least $k = \lceil \log_2(1/\epsilon) - 1 \rceil$ questions in order to find x_ϵ. This bound can be achieved by asking bisection questions, i.e., by asking whether x is less than the midpoint α of an interval of uncertainty. For the first question $\alpha = \frac{1}{2}$, and for the second α is either $\frac{1}{4}$ or $\frac{3}{4}$, depending on the answer to the first question. Then x_ϵ is the midpoint of the last interval of uncertainty. Note that in order to pose the ith bisection question we have to know the answers to the $(i - 1)$ previous questions.

We can find x_ϵ by asking k simultaneously posed questions where k is as

before. The ith question is given by: "Is the ith bit of x zero?" for $i = 1, \ldots, k$. Then we set

$$x_\epsilon = \sum_{i=1}^{k} b_i 2^{-i} + 2^{-(k+1)}$$

here $b_i = 0$ if the ith question is answered affirmatively and $b_i = 1$ otherwise. When all b_i, $i = 1, \ldots, k$, are known, the number x_ϵ can be obtained with no extra cost. Indeed, b_i is the ith bit of x_ϵ, i.e., $x_\epsilon = .b_1 b_2 \cdots b_k 1$ in binary notation.

From this we easily conclude that the ϵ-complexity is given by

$$\operatorname{comp}(\epsilon) = c\lceil \log_2 1/\epsilon - 1 \rceil.$$

This means that no matter which questions are asked and how the answers are combined, it is impossible to find x_ϵ with cost less than $\operatorname{comp}(\epsilon)$. Questions about successive bits of x and the rule of combining them given by $x_\epsilon = .b_1 b_2 \cdots b_k 1$ are *optimal* in the sense of minimizing the cost.

We wish to stress that $\operatorname{comp}(\epsilon)$ denotes the *worst case* ϵ-complexity. That is, the error between x and x_ϵ is *always* not greater than ϵ, and the cost of obtaining x_ϵ is defined by the worst case.

Continuous binary search for the *average case* is analyzed in Section 5. Here, average case means that the error and the cost are defined on the average relative to some probability measure.

Continuous binary search with *contaminated information* is analyzed by Rivest, Meyer, Kleitman, Winklmann, and Spencer (1980). They seek a subset A of $F = (0, 1)$ such that $x \in A$ and Lebesgue measure of A does not exceed ϵ. They assume that up to p of the questions receive erroneous answers. The result is that for small ϵ, roughly $\log_2 \epsilon^{-1} + p \log_2 \log_2 \epsilon^{-1}$ questions are needed to solve the problem.

EXAMPLE 1.2 (Linear Equations). We wish to approximate the solution of a large system of linear equations $Ax = b$. Here A is an $n \times n$ nonsingular matrix and b is a vector. Without loss of generality we can assume b is normalized so that $\|b\|_2 = 1$. Assume that n is so large that the $O(n^2)$ storage and/or $O(n^3)$ arithmetic operations needed for standard direct methods are not feasible. The matrix A is often sparse, which means that one can supply a subroutine which computes Az for any vector z in time and storage proportional to n. Typically A enjoys certain properties. That is, $A \in F$, where F is a class of nonsingular $n \times n$ matrices. Examples of such classes F include symmetric matrices, symmetric positive definite matrices, and matrices with bounded condition number.

Using this type of information we want to find an approximate solution with minimal cost. The linear equations problem is formalized as follows:

Problem. For each A in F, compute a vector x_ϵ such that $\|Ax_\epsilon - b\|_2 \le \epsilon$, $\epsilon \in (0, 1)$.

Information. (i) We know a priori that $A \in F$ and we know the vector b, $\|b\|_2 = 1$.

(ii) We can *only* compute Az for any vector z.

Model of computation. (i) Each matrix–vector multiplication Az costs c, $c > 0$.

(ii) We can perform arithmetic operations, comparisons, and the evaluation of elementary functions (such as square root, sine, cosine, and the like) in the real number model. Each of them costs unity.

Let $\mathrm{comp}(\epsilon)$ denote the ϵ-complexity of this problem, i.e., the minimal cost of computing $x_\epsilon = x_\epsilon(A, b)$. Here, as in Example 1.1, we assume a worst case setting. That is, the error $\|Ax_\epsilon(A, b) - b\|_2$ has to be no greater than ϵ for *all* matrices A from the class F and the cost of computing $x_\epsilon(A, b)$ is defined by a worst A.

What is the ϵ-complexity of the linear equations problem? Which matrix–vector multiplications should be performed and how should the obtained vectors be combined in order to compute x_ϵ with minimal cost?

The answer depends on the class F. We report here the result when F consists of symmetric positive definite matrices with uniformly bounded condition numbers. That is,

$$F = \{A : A = A^{\mathrm{T}} > 0,\ \|A\|_2\,\|A^{-1}\|_2 \le M\}.$$

On the basis of the results of Nemirovsky and Yudin (1983, p. 262) and Traub and Woźniakowski (1984b) one can show that for small ϵ, large M, and $n > M^{1/2}\ln(2/\epsilon)/2$, we have

$$\mathrm{comp}(\epsilon) = (c + a_\epsilon)M^{1/2}\ln(2/\epsilon)/4,$$

where $a_\epsilon \in [0, c + 10n]$.

The upper bound is obtained if the vector x_ϵ is computed by the minimal residual algorithm using Krylov information $N_k(A, b) = [b, Ab, \ldots, A^k b]$ with k roughly equal to $M^{1/2}\ln(2/\epsilon)/2$. Thus, Krylov information and the minimal residual algorithm are almost *optimal* in the sense of minimizing the cost.

A detailed analysis of this problem for different classes F is considered by Chou (1985). He also analyzes the complexity of the approximate solution of the eigenvalue problem using the results of Nemirovsky and Yudin (1983) and Kuczyński (1985).

The average case complexity of the linear equations problem (as well as of the eigenvalue problem) has not yet been studied.

EXAMPLE 1.3 (Integration). Suppose we wish to approximate the integral $\int_0^{2\pi} f(t)\,dt$ of a scalar real function f for $f \in F$. Assume we know some global properties of f, e.g., F is a class of smooth, convex, or periodic functions. We also have a subroutine which computes $f(t)$ for any point t from the interval $[0, 2\pi]$. We are charged for each subroutine call.

Using this type of information we want to compute an approximate value of the integral with minimal cost. The integration problem is formalized as follows:

Problem. For each f in F, compute a real number x_ϵ such that $\left|\int_0^{2\pi} f(t)\,dt - x_\epsilon\right| \le \epsilon$, $\epsilon > 0$.

Information. (i) We know that $f \in F$.

(ii) We can compute $f(t)$ for any $t \in [0, 2\pi]$.

Model of Computation. (i) Each function evaluation costs c, $c > 0$.

(ii) We can perform arithmetic operations, comparisons, and the evaluation of elementary functions in the real number model. Each of them costs unity.

Let $\mathrm{comp}(\epsilon)$ denote the ϵ-complexity of the of the integration problem, i.e., the minimal cost of computing $x_\epsilon = x_\epsilon(f)$ for the worst case. Thus, the error $\left|\int_0^{2\pi} f(t)\,dt - x_\epsilon(f)\right|$ has to be no greater than ϵ for *all* f from the class F and the cost of computing $x_\epsilon(f)$ is defined by a worst f.

What is the ϵ-complexity of the integration problem? At which points should f be sampled and how should these samples be combined in order to compute x_ϵ with minimal cost?

We report here the result for one particular class F consisting of periodic functions whose $(r-1)$st derivative is absolutely continuous and whose rth derivative is bounded in the L_∞ norm by unity, $r \ge 1$,

$$F = F_r = \{f\colon [0, 2\pi] \to \mathbf{R} : f \text{ is periodic, } f^{(r-1)} \text{ is absolutely continuous, } \|f^{(r)}\|_\infty \le 1\}.$$

On the basis of the results of Bakhvalov (1971) and Motornyj (1973) we have

$$\mathrm{comp}(\epsilon) = (c+1)\lceil (2\pi K_r/\epsilon)^{1/r}\rceil + a_\epsilon.$$

Here K_r is the Favard constant,

$$K_r = 4/\pi \sum_{i=0}^{\infty} (-1)^{i(r+1)}/(2i+1)^{r+1}, \qquad K_r \in [1, \pi/2],$$

and $a_\epsilon = -1$ or $a_\epsilon = 0$.

Note that we know the ϵ-complexity to within the cost of one arithmetic

operation. Let $n = \lceil (2\pi K_r/\epsilon)^{1/r} \rceil$. The approximate value of the integral can be computed by the composite midpoint rule

$$x_\epsilon = 2\pi/n \sum_{i=1}^{n} f(2\pi(i-1)/n).$$

Thus, the sampling of f at equally spaced points and using the very simple composite midpoint rule turn out to be *optimal* for the class F. We stress that this holds only because of the periodicity of the functions f. For many different classes F, different and more complicated formulas are optimal. The reader interested in the integration problem is referred to Traub and Woźniakowski (1980), where about 100 papers are cited with optimal formulas and complexity results for many different classes F for the worst case setting.

The average case setting for the integration problem is studied, for instance, in a recent paper of Lee and Wasilkowski (1985).

We hope that the three examples presented above give the reader some intuition about information-based complexity. Each problem of information-based complexity consists of three concepts:

—problem formulation,
—information,
—model of computation.

We now briefly discuss these three concepts.

1. *The problem formulation* states what we want to approximate, for which problem elements we are seeking this approximation, and what we mean by error criterion. For instance, for the integration problem we want to approximate the integral, problem elements are functions from a given class, and the error criterion is that the absolute difference between the integral and its approximation does not exceed ϵ for any integrand from the given class.

2. *Information* is described by

—a priori knowledge that the problem elements belong to a set. Examples of such sets are: the set of real numbers on the open unit interval, the set of symmetric matrices, or the set of functions of a prescribed smoothness.

—certain information operations on problem elements. Examples of such operations are: questions, matrix–vector multiplications, or samplings of a function.

The three fundamental assumptions on information are:

—Information is *partial*. That is, having a priori knowledge and a finite number of information operations, we cannot, in general, identify the problem element exactly. Therefore we cannot solve the problem exactly.

—Information is *contaminated*. That is, it is computed with some error. Examples of such error are: some questions receive erroneous answers; matrix–vector multiplications and function evaluations are computed with

round-off. (For simplicity, we defer discussion of information error to Section 5.)

—Information is *priced*. That is, we are charged for each information operation on a problem element.

3. *The model of computation* states what we are charged for *permissible information operations* and for *permissible combinatory operations* on this information.

Remark 1.1. In the three examples we assume the real number model of computation. That is, we assume that

—real numbers are used;

—arithmetic operations, comparisons, and the evaulation of elementary functions are performed exactly (i.e., with infinite precision) with unit cost.

Although infinite precision does not exist in actual computation, it is a very useful mathematical abstraction. Further discussion on the real number model and on other models of computation may be found in Section 5. We emphasize that the "user" can choose any model he likes such as various formal models of theoretical computer science.

We are now ready to discuss the concept of ϵ-complexity which is central to information-based complexity. The ϵ-complexity is the *minimal cost* which is required to compute approximations to within ϵ. To do this, we use information operations and an algorithm which obtains an approximation by performing combinatory operations on the information. The central issue of information-based complexity is to find information and an algorithm which are *optimal;* i.e., they compute approximations to within ϵ with cost equal to the ϵ-complexity.

Usually we can only find bounds on ϵ-complexity. To obtain an upper bound, it is enough to show that a certain number of information operations and a number of combinatory operations are sufficient to compute an approxiomation x_ϵ. The total cost of such operations is then an upper bound on the ϵ-complexity. To obtain a lower bound on the ϵ-complexity one has to analyze *all* possible ways by which an approximation x_ϵ can be computed. For example, we give a lower bound on the ϵ-complexity for the linear equations problem for a certain class of matrices. This means that, no matter which matrix–vector multiplications are performed and how these vectors are combined, it is impossible to find an approximation x_ϵ with cost less than the lower bound.

Typically, in computational complexity it is hard to find good lower bounds. We stress that in information-based complexity, lower bounds are often found by analyzing permissible information. We find the minimal number of information operations needed to identify the exact solution to within a prescribed accuracy.

In the three examples we consider the *worst case* setting. The worst case setting is characterized by two properties:

—the error is no greater than a prescribed accuracy for *all* problem elements;

—the cost of computing an approximation is defined by a *worst* problem element.

A different approach might be to define the error and cost on the *average*. This leads to an average case setting in information-based complexity. This setting is briefly discussed in Section 4. A survey of average case may be found in Wasilkowski (1985a). There are also some other approaches such as probabilistic or asymptotic settings. They are briefly mentioned in Section 5.

We summarize the rest of this paper. Section 2 generalizes the examples of the Introduction. It deals with a normed worst case setting. The notion of information is precisely defined. Nonadaptive (parallel) and adaptive (sequential) information are discussed. We show that the intrinsic uncertainty of information is measured by its *radius of information*. The radius of information plays a major role in our study. The notions of ϵ-complexity and optimality of information and algorithm are precisely defined.

Section 3 deals with *linear* problems. Results concerning the power of adaption and the existence of linear optimal error algorithms are addressed. Spline algorithms, optimal information, and ϵ-complexity of linear problems are briefly discussed. In particular, we indicate that the ϵ-complexity of a linear problem can be arbitrarily large. A list of linear problems of special interest and some nonlinear problems studied from the information-based complexity point of view conclude Section 3.

Sections 1 through 3 are primarily devoted to the worst case setting of information-based complexity. In Section 4 we briefly discuss the average case setting and illustrate it by the continuous binary search problem. This analysis seems to be new. Section 5 completes the paper by outlining alternate settings of information-based complexity and indicating future research directions.

2. Normed Worst Case Setting

2.1. *Formulation*

We present an abstract formulation which includes numerous important problems as special cases. This is the *normed worst case* setting. It is based on two major assumptions:

—uncertainty is measured by a norm;

—a worst case is used to define the error and the cost of algorithms.

Let F and G be given sets and let S be a given mapping

$$S: F \to G.$$

We call S the *solution operator*. The set G is assumed to be a subset of a normed linear space. For given nonnegative ϵ, our goal is to compute an *ϵ-approximation* $x_\epsilon = x_\epsilon(f)$ to the element $S(f), f \in F$, i.e., to compute x_ϵ such that

$$\|S(f) - x_\epsilon\| \leq \epsilon.$$

In order to compute x_ϵ we must know something about f. Assume that for $f \in F$ we can gather knowledge about f by computing some $L(f)$. Here $L: F \to H$ for some set H. Let Λ denote a class of permissible information operations L.

For example, for the continuous binary search problem, Λ consists of questions, i.e., $L(f) \in H = \{0, 1\}$. For the linear equations problem, Λ consists of matrix–vector multiplications, i.e., for $f = A$, $L(A) = Az$ for some vector z, $H = \mathbf{R}^n$. For the integration problem, Λ consists of function evaluations, i.e., $L(f) = f(z)$for some point z, $H = \mathbf{R}$.

We are charged for each computation of $L(f)$. We assume that we can also perform certain operations, such as the addition of two elements of G, $g_1 + g_2$ for $g_i \in G$, multiplication by scalars, αg for $\alpha \in \mathbf{R}$ and $g \in G$. We assume here the real number model as in the examples of the Introduction. Thus, we assume that each operation is exactly performed with unit cost. We wish to find an ϵ-approximation with minimal cost. The normed worst case setting is formalized as follows:

Problem. For each f in F, compute x_ϵ such that $\|S(f) - x_\epsilon\| \leq \epsilon, \epsilon > 0$.

Information. (i) We know a priori the solution operator S and that $f \in F$.
(ii) We can compute $L(f)$ for any $L \in \Lambda$ and any $f \in F$.

Model of computation. (i) Each information operation $L(f)$ costs c.
(ii) We can perform certain combinatory operations exactly, such as addition of two elements from G and multiplication by scalars, at unit cost.

Remark 2.1. We stress that many (but not all) problems can be formulated in the normed setting presented above. For instance, the continuous binary search problem corresponds to $F = G = (0, 1)$ and $S(f) = f$; the integration problem corresponds to F being a class of real functions, $G = \mathbf{R}$ and $S(f)$ is the integral of f over the interval $[0, 2\pi]$.

What about the linear equations problem? It is easy to see that it cannot be formulated in this setting. It can be formulated if we measure uncertainty differently. We achieve this as follows.

Let F and G be given sets and let W be a given mapping

$$W: F \times \mathbf{R}_+ \to 2^G,$$

where $\mathbf{R}_+ = [0, +\infty)$ and 2^G is the class of all subset of G. Thus, $W(f, \epsilon)$ is a subset of G.

We assume that $W(f, \epsilon)$ is nonempty and grows as ϵ increases. Thus the mapping W has two properties

(i) $W(f, 0) \neq \emptyset$, $\forall f = F$,
(ii) $\epsilon_1 \leq \epsilon_2$ implies $W(f, \epsilon_1) \subseteq W(f, \epsilon_2)$, $\forall \epsilon_1, \epsilon_2 \in \mathbf{R}_+$ and $\forall f \in F$.

These two properties of W enable us to define an ϵ-approximation. An *ϵ-approximation of f* is now an element x_ϵ of G such that

$$x_\epsilon \in W(f, \epsilon).$$

How restrictive are the assumptions (i) and (ii)? The first one states that there is something to find. With the interpretation that ϵ measures uncertainty, the second assumption states that as the required uncertainty decreases, the set of elements that satisfies the criterion becomes smaller. Thus, one may view these two properties as nonrestrictive.

The mapping W is called a *generalized solution operator*. It clearly generalizes the previous definition since $W(f, \epsilon) = \{x \in G : \|S(f) - x\| \leq \epsilon\}$ satisfies (i) and (ii). The linear equations problem is now formulated as

$$W(A, \epsilon) = \{x \in \mathbf{R}^n : \|Ax - b\| \leq \epsilon\},$$

where $f = A$ belongs to F and $G = \mathbf{R}^n$.

We do not pursue the analysis of the generalized operator W here. The reader can find such an analysis in Traub, Wasilkowski, and Woźniakowski (1983). A relation between a normed setting and this generalized setting may be found in Werschulz (1983a).

2.2. *Information*

How can we gather knowledge about a problem element f in order to compute an ϵ-approximation? We can compute $L(f)$ for any operation L, $L: F \to H$, from the class Λ. Suppose we decide to perform $n(f)$ such operations on f. Clearly, since we are charged for each of them, we would like to minimize the number $n(f)$ of operations which are necessary in order to find an ϵ-approximation.

We now discuss two classes of information. The first one is the class of *nonadaptive* information. Namely, N is called *nonadaptive* information iff

$$N(f) = [L_1(f), L_2(f), \ldots, L_n(f)], \qquad L_i \in \Lambda.$$

That is, for any f we perform the same number of operations, $n(f) \equiv n$, and they are given by permissible L_i. Note that $L_1, L_2, \ldots, L_n$ are given simultaneously. The number n is called the *cardinality* of information N. Sometimes we write $N = N^{\text{non}}$ to stress that N is nonadaptive.

The computation of nonadaptive information can be done in parallel very efficiently. Indeed, if one has n processors then the ith processor can compute $L_i(f)$, $i = 1, 2, \ldots, n$, and the total time needed to compute $N(f)$ is equal to the maximal time needed by one of the n processors. That is why nonadaptive information is sometimes called *parallel* information.

The second class of information is called *adaptive*. For adaptive information the number $n(f)$ of operations may vary with an element f and the choice of ith operation L_i may depend on the $(i - 1)$ previously computed values. More precisely, N is called *adaptive* if

$$N(f) = [L_1(f), L_2(f; y_1), \ldots, L_{n(f)}(f; y_1, \ldots, y_{n(f)-1})],$$

where $y_1 = L_1(f)$ and $y_i = L_i(f; y_1, \ldots, y_{i-1})$ for $i = 2, 3, \ldots, n(f)$. Thus, y_i denotes the ith value of information operation. Here we assume that for fixed $y_1, \ldots, y_{i-1}$, $L_i(\cdot; y_1, \ldots, y_{i-1}) \in \Lambda$. The number $n(f)$ denotes the total number of operations involving the element f, and is called the *cardinality of* N at f.

The number $n(f)$ is determined as follows (see also Wasilkowski, 1985b). Suppose that we have already computed $y_1 = L_1(f)$, $y_2 = L_2(f; y_1), \ldots,$ $y_i = L_i(f; y_1, \ldots, y_{i-1})$. Then we make a decision whether another information operation is needed. The decision is made based on available knowledge about f. That is, we have a Boolean function ter_i: $H^i \to \{0, 1\}$, called a *termination function*. If $\text{ter}_i(y_1, y_2, \ldots, y_i) = 1$ then we terminate the computation and $n(f) = i$. Otherwise, if $\text{ter}_i(y_1, y_2, \ldots, y_i) = 0$, we choose the $(i + 1)$st permissible operation $L_{i+1}(\cdot; y_1, \ldots, y_i)$ and compute $L_{i+1}(f; y_1, \ldots, y_i)$. This process is then repeated. Thus, the cardinality $n(f)$ at f is defined as

$$n(f) = \min\{i : \text{ter}_i(y_1, y_2, \ldots, y_i) = 1\},$$

where $y_1 = L_1(f)$, $y_2 = L_2(f; y_1)$ and so on. Although we do not need to assume that $n(f)$ is finite, we usually choose termination functions in such a way that $n(f)$ is finite. This can be done, for instance by taking $\text{ter}_k(y_1, \ldots, y_k) \equiv 1$ for a large k.

By the *cardinality* of adaptive information N we mean

$$n = \sup\{n(f) : f \in F\}.$$

Sometimes we write $N = N^a$ to stress that N is adaptive.

Adaptive information requires sequential computation. We have to wait until y_i is computed in order to decide whether another information operation is needed, and if so, what the $(i + 1)$st permissible operation is. That is why adaptive information is sometimes called *sequential* information.

To illustrate nonadaptive and adaptive information consider f as a scalar function and

$$N(f) = [f(t_1), f(t_2), \ldots, f(t_{n(f)})].$$

If $n(f) \equiv n$ and the points t_i are given simultaneously (a priori) then N is nonadaptive. If $n(f)$ varies and/or the choice of the point t_i depends on $f(t_i)$, $f(t_2), \ldots, f(t_{i-1})$ then N is adaptive.

In either case, knowing $N(f)$ we are, in general, unable to identify f uniquely. The information operator N is many-to-one and there exist many elements f which share the same information. The information is called *partial* since $N(f)$ supplies only partial knowledge about f.

Note also that the computation of $N(f)$ *costs*. It is clear that the cost of $N(f)$ depends on the cardinality $N(f)$.

2.3. *Radius of Information*

Let N be adaptive or nonadaptive information as defined in Section 2.2. Since N is partial, it causes uncertainty in the solution. Let $y = N(f)$ be the computed information about f. Then

$$N^{-1}(y) = \{\tilde{f} \in F : N(\tilde{f}) = y\}$$

is the set of indistinguishable problem elements, and

$$SN^{-1}(y) = \{S(\tilde{f}) \in G : \tilde{f} \in F, N(\tilde{f}) = y\}$$

is the set of indistinguishable solution elements. This can be schematized as in Fig. 2.1.

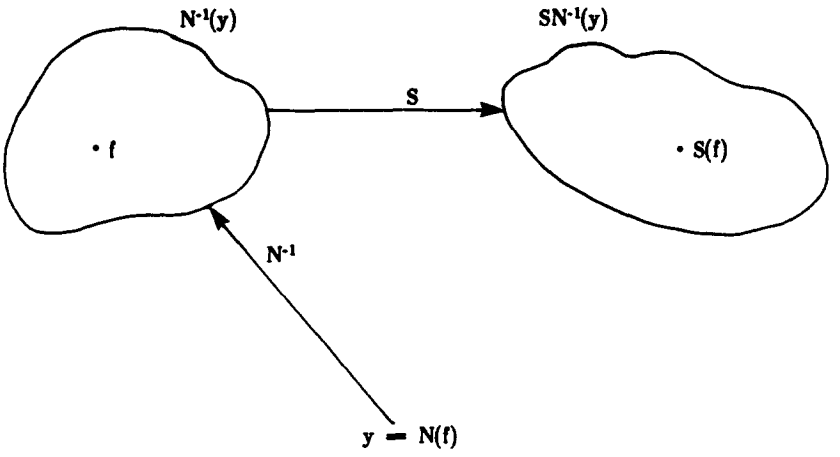

FIG. 2.1

We want to compute x_ϵ based on $y = N(f)$. That is, $x_\epsilon(N(f))$. The element x_ϵ should approximate $S(f)$ as well as all elements $S(f)$ for $f \in N^{-1}(y)$. We can guarantee that x_ϵ is a good approximation iff the set $SN^{-1}(y)$ is "small." The smallness of the set $SN^{-1}(y)$ is defined by its *radius*.

For a given set A in a normed linear space G, the radius of A is defined by

$$\operatorname{rad}(A) = \inf_{x \in G} \sup_{a \in A} \|x - a\|.$$

This is, roughly speaking, the radius of the smallest ball which contains the set A. Note that $\operatorname{rad}(SN^{-1}(y))$ is the radius of the set of indistinguishable solution elements. It is clear that we can find an ϵ-approximation iff $\operatorname{rad}(SN^{-1}(y)) \leq \epsilon$ (modulo a technical assumption that the infimum is attained). Define the (global) *radius of information* as

$$r(N) = \sup_{y \in N(F)} \operatorname{rad}(SN^{-1}(y)).$$

The radius of information is the radius of the smallest ball which contains $SN^{-1}(y)$ for a worst y. Thus, we have

THEOREM. *We can find an ϵ-approximation for all $f \in F$ iff*

$$r(N) \leq \epsilon.$$

The radius of information plays a major role in information-based complexity. It measures the uncertainty due to partial information. Observe that $r(N)$ depends on the solution operator S and on the information N. It does not depend on how an approximation x_ϵ is constructed.

Often, as the cardinality of information goes to infinity, the radius goes to zero. Then we can find the minimal cardinality for which the radius does not exceed ϵ. Since the cost of computing information depends on the cardinality, the ϵ-complexity is bounded below by a quantity which is proportional to the minimal cardinality. This will be discussed in Section 2.5.

The radius of information can be defined for every setting. For the average case setting the reader is referred to Wasilkowski (1985a).

2.4. *Algorithms*

Let $N(f)$ be the computed information. Knowing $N(f)$, an approximation x_ϵ is computed. Thus, $x_\epsilon = \phi(N(f))$, where ϕ is a mapping,

$$\phi: N(F) \to G.$$

We call ϕ an (idealized) *algorithm*. The algorithm ϕ combines the known information (input) and produces an approximation (output) to $S(f)$. An idealized algorithm is *any* rule using the information $N(f)$. We stress that this is a very general notion of algorithm. For some complicated mappings ϕ,

implementation may not be possible. On the other hand, our model of computation tells what can be computed and how much it costs. Recall that we wish to compute x_ϵ with minimal cost. If the cost of $\phi(N(f))$ is high, ϕ will be automatically eliminated as one whose cost is far from being minimal.

Of course, we sometimes want to guarantee that an algorithm ϕ has some additional properties such as being on-line or enjoying numerical stability. Then we should restrict the class of idealized algorithms to a class of "realizable" algorithms. We emphasize that restricting the notion of algorithm can only decrease the quality of approximations.

Let ϕ be an (idealized) algorithm. We define the (worst case) *error* of an algorithm ϕ,

$$e(\phi, N) = \sup\{\|S(f) - \phi(N(f))\| : f \in F\},$$

as the maximal distance between the solution element $S(f)$ and the approximation $\phi(N(f))$ computed by the algorithm ϕ. From Section 2.3, we immediately conclude

THEOREM. *The radius of information is a sharp lower bound on the error of any algorithm using N. Thus*

$$r(N) = \inf_{\phi} e(\phi, N).$$

This theorem is very useful. It enables us to concentrate on the information level without the need of analyzing algorithms as far as uncertainty is concerned.

An algorithm ϕ that uses N and whose error is minimal, $e(\phi, N) = r(N)$, is called an *optimal error algorithm*.

For the average case setting, the error of an algorithm ϕ is defined by the expected value of $\|S(f) - \phi(N(f))\|$ with respect to a given probability measure on F. The average radius of information is also a sharp lower bound on the average error of algorithms (see Wasilkowski, 1983).

2.5. *Complexity and Optimality*

In this section we precisely define the notions of ϵ-complexity and optimality of information and algorithm. Recall that the approximation $x_\epsilon = \phi(N(f))$ is computed in two steps. First $y = N(f)$ is computed, and then $x_\epsilon = \phi(y)$ is computed. Let $\text{cost}(N, f)$ denote the information cost of computing $N(f)$. We are charged for the computation of $L_i(f)$, $i = 1, \ldots, n(f)$, and if N is adaptive, we are charged for the selection of L_i. Using our model of computation, $\text{cost}(N, f) \geq cn\,(f)$, where $n(f)$ is the cardinality of N at f. If N is nonadaptive, the operations L_i are given a priori and $\text{cost}(N, f) = cn\,(f)$. Let $\text{cost}(\phi, y)$, $y = N(f)$, denote the combinatory cost

of computing $\phi(y)$. That is, if the computation of $\phi(y)$ requires the evaluation of k combinatory operations, then $\text{cost}(\phi, y) = k$.

Define the (worst case) *cost* of the algorithm ϕ using information N as

$$\text{cost}(\phi, N) = \sup\{\text{cost}(N, f) + \text{cost}(\phi, N(f)) : f \in F\}.$$

Thus, $\text{cost}(\phi, N)$ is the cost of computing $y = N(f)$ and $\phi(y)$ for a worst f.

The *ϵ-complexity* is defined as the minimal cost of computing an ϵ-approximation,

$$\text{comp}(\epsilon) = \inf\{\text{cost}(\phi, N) : \phi, N \quad \text{such that} \quad e(\phi, N) \leq \epsilon\}.$$

We stress once more that $\text{comp}(\epsilon)$ is defined as the ϵ-complexity for the worst case since both the error and the cost are defined by worst performances.

Information N, and an algorithm ϕ that uses N, for which

$$\text{cost}(\phi, N) = \text{comp}(\epsilon) \qquad \text{and} \qquad e(\phi, N) \leq \epsilon$$

are called *optimal information* and an *optimal algorithm*, respectively. More precisely, N should be called optimal ϵ-complexity information and ϕ an optimal ϵ-complexity algorithm. Since ϵ is regarded as fixed, we simplify the terminology by dropping the word ϵ-complexity. Thus, an optimal algorithm using optimal information computes an ϵ-approximation with minimal cost. The central issue of information-based complexity is to find the ϵ-complexity as well as optimal information and an optimal algorithm.

For the average case, the cost of an algorithm is defined by the expected value of $\text{cost}(N, f) + \text{cost}(\phi, N(f))$ with respect to a given probability measure on F. Then the average ϵ-complexity and optimality of information and algorithm are defined as above with average errors and costs respectively (see Wasilkowski 1985a).

2.6. *Cardinality Number*

We now show how bounds on the ϵ-complexity can be derived using the notion of radius of information. Recall that the radius $r(N)$ of information measures the intrinsic uncertainty of partial information. Let $\text{card}(N)$ denote the cardinality of information, i.e., the total number of information operations in $N(f)$ for a worst f.

By the *ϵ-cardinality number* $m(\epsilon)$ we mean the minimal cardinality of information whose radius does not exceed ϵ,

$$m(\epsilon) = \min\{\text{card}(N) : r(N) \leq \epsilon\}$$

In order to compute an ϵ-approximation we have to use information N whose cardinality is at least $m(\epsilon)$. Furthermore the cost of any algorithm ϕ

using such information has to be at least $cm(\epsilon)$. Therefore we have the following lower bound on the ϵ-complexity,

$$\text{comp}(\epsilon) \geq cm(\epsilon).$$

Suppose that N is information such that

(i) its cardinality is $m(\epsilon)$,
(ii) its radius is at most ϵ,
(iii) its information cost is $cm(\epsilon)$.

Let ϕ be an algorithm that uses N and whose error $e(\phi, N)$ is equal to $r(N)$. That is, ϕ computes an ϵ-approximation. Assume that the combinatory cost of ϕ is dominated by the information cost,

$$\text{cost}(\phi, N(f)) \ll \text{cost}(N, f), \qquad \forall f \in F.$$

Then we have $\text{cost}(\phi, N) \sim cm(\epsilon)$. From the above inequality, we conclude that N and ϕ are almost *optimal information and algorithm*, and

$$\text{comp}(\epsilon) \sim cm(\epsilon).$$

Thus, we have found the ϵ-complexity almost exactly and it is exhibited in terms of the ϵ-cardinality number. We stress that this holds whenever there exist information which satisfies assumptions (i), (ii), (iii) and an optimal error algorithm for which the information cost dominates the combinatory cost.

How restrictive are these assumptions in practice? Surprisingly enough, these assumptions hold for many important problems. They hold for the three examples mentioned in the Introduction. They also hold for many linear problems, as will be discussed in Section 3.

There are some counterexamples as well. One is due to Papadimitriou and Tsitsiklis (1984). They present a nonlinear problem of decentralized control theory for which $m(\epsilon) = \Theta((1/\epsilon)^4)$ and for which the combinatory cost is not polynomial in $1/\epsilon$ iff the famous conjecture P $\neq$ NP holds. Thus, it is very likely that for this problem, the combinatory cost dominates the information cost and $\text{comp}(\epsilon)$ is much greater than the ϵ-cardinality.

Also Nemirovsky and Yudin (1983) indicate a number of minimization problems for which there are no known algorithms with combinatory cost comparable to the information cost and whose errors are proportional to the radius of information.

3. Linear Problems

3.1. *Definition and Brief History*

In this section we discuss some of the major results for linear problems. We

begin with an example of a linear problem. As in Example 3.1, let

$$S(f) = \int_0^{2\pi} f(t)\, dt.$$

We want to approximate $S(f)$ knowing

$$N(f) = [f(t_1), \ldots, f(t_n)].$$

In order that the ϵ-complexity be finite, one has to restrict the class F of integrands. Assume that F consists of functions whose rth derivatives are uniformly bounded. Without loss of generality, assume $|f^{(r)}(t)| \leq 1$ for any t.

This example is a special case of the following formulation. Let F be a subset of a linear space F_1 having the form

$$F = \{f \in F_1 : \|Tf\| \leq 1\},$$

where T: $F_1 \to X$ is a *linear* mapping into a normed linear space X.

Note that F is balanced, $f \in F$ implies $-f \in F$, and convex, $f, g \in F$ and $t \in [0, 1]$ implies $tf + (1 - t)g \in F$. We will use these properties of F below.

Let G be a normed linear space and let the solution operator S, S: $F_1 \to G$, be a *linear* mapping. We approximate $S(f)$ for $f \in F$ by using *linear* functionals as permissible information operations. That is, Λ is a class of linear functionals L, L: $F_1 \to \mathbf{R}$.

By a *linear problem* we mean S, F, and Λ defined as above.

We give a brief historical note on the study of linear problems mentioning only the most important papers. Sard (1949) studied optimal linear algorithms for the integration problem which use function evaluations at fixed points. Independently, Nikolskij (1950) posed the same problem and permitted the evaluation points to be optimally chosen. Golomb and Weinberger (1959) performed the first systematic study of optimal error algorithms for the approximation of a linear functional. Micchelli and Rivlin (1977) studied optimal error algorithms for linear operators using linear contaminated information. The study of the complexity of problems with partial, contaminated, and priced information is initiated in Traub and Woźniakowski (1980) and Traub, Wasilkowski, and Woźniakowski (1983). Substantial portions of these two monographs are devoted to the complexity of linear problems.

3.2. *Adaption*

Is adaptive information more powerful than nonadaptive information for linear problems? In this section we show that adaption does *not* help for linear problems for the worst case setting. We then briefly discuss the same issue for different settings.

Our interest in the power of nonadaptive information is motivated by a number of reasons:

1. For nonadaptive information we have a natural decomposition for parallel computation. Because nonadaptive information minimizes communication requirements, it is also desirable for distributed computation.

2. Nonadaptive information is much simpler and therefore much easier to analyze than adaptive information. If we know that nothing can be gained by using adaptive information, we significantly cut the search space when seeking which information should be used to minimize the cost of computing an ϵ-approximation.

Let N^a be adaptive information. That is,

$$N^a(f) = [L_1(f), L_2(f; y_1), \ldots, L_{n(f)}(f; y_1, \ldots, y_{n(f)-1})],$$

where $y_1 = L_1(f)$, $y_i = L_i(f; y_1, \ldots, y_{i-1})$ and the cardinality $n(f) = \min\{i : \mathrm{ter}_i(y_1, \ldots, y_i) = 1\}$ for some Boolean functions $\mathrm{ter}_i: \mathbf{R}^i \to \{0, 1\}$. Here $L_i(\cdot; y_1, \ldots, y_i)$ is a linear functional from the given class Λ.

For given adaptive information N^a, we define nonadaptive information N^{non} of cardinality $n(0)$ by fixing all values y_i to zero. (Note that $f = 0 \in F$ and therefore $n(0)$ is well defined.) That is,

$$N^{non}(f) = [L_1(f), L_2(f; 0), \ldots, L_{n(0)}(f; 0, \ldots, 0)].$$

Clearly $\mathrm{card}(N^{non}) \le \mathrm{card}(N^a)$. Information N^{non} is nonadaptive since $L_i(\cdot; 0, \ldots, 0)$ are given simultaneously and the same number of evaluations is performed for each f. Observe that the structure of N^{non} is simpler than the structure of N^a and that $L_i(\cdot; 0, \ldots, 0) \in \Lambda$.

We compare the power of adaptive information N^a with the power of nonadaptive information N^{non} by their radii of information, i.e., by the intrinsic uncertainty caused by N^a and N^{non}, respectively. We have

THEOREM. $r(N^{non}) \le 2\, r(N^a)$.

Proof. We provide a proof because it is short and straightforward. Note that N^{non} is a linear mapping and therefore for $y = N^{non}(f)$, $f \in F$, we have

$$S(N^{non})^{-1}(y) = \{Sf \in G : \tilde{f} \in F,\ N^{non}(\tilde{f} - f) = 0\}.$$

For any set A, $A \subseteq G$, the radius of A is related to its diameter

$$\mathrm{diam}(A) = \sup\{\|a_1 - a_2\| : a_1, a_2 \in A\}$$

by

$$\mathrm{rad}(A) \le \mathrm{diam}(A) \le 2\, \mathrm{rad}(A).$$

Then linearity of S yields

$$\begin{aligned}\operatorname{diam}(S(N^{\text{non}})^{-1}(y)) &= \sup\{\|S(f_1 - f_2)\| : f_i \in F,\ N^{\text{non}}(f_i - f) = 0,\ i = 1, 2\} \\ &\le \sup\{\|S(f_1 - f_2)\| : (f_1 - f_2)/2 \in F,\ N^{\text{non}}(f_1 - f_2) = 0\} \\ &= 2 \sup\{\|Sh\| : h \in F,\ N^{\text{non}}(h) = 0\}.\end{aligned}$$

Here we used the fact that F is convex and balanced and therefore $f_1, f_2 \in F$ implied that $(f_1 - f_2)/2 \in F$.

For $f = 0$ we have $y = 0$ and the above inequality becomes the equality. Thus

$$\operatorname{diam}(S(N^{\text{non}})^{-1}(0)) \ge \operatorname{diam}(S(N^{\text{non}})^{-1}(y)) \ge \operatorname{rad}(S(N^{\text{non}})^{-1}(y)).$$

Since this holds for all y, we have

$$\operatorname{diam}(S(N^{\text{non}})^{-1}(0)) \ge r(N^{\text{non}}).$$

On the other hand, for $f = 0$ the construction of N^{non} yields that the sets $(N^{\text{a}})^{-1}(0)$ and $(N^{\text{non}})^{-1}(0)$ are the same. Thus

$$\begin{aligned}r(N^{\text{a}}) \ge \operatorname{rad}(S(N^{\text{a}})^{-1}(0)) &= \operatorname{rad}(S(N^{\text{non}})^{-1}(0)) \\ &\ge \operatorname{diam}(S(N^{\text{non}})^{-1}(0))/2 \ge r(N^{\text{non}})/2.\end{aligned}$$

This completes the proof. ■

The factor 2 in the theorem is not needed for many linear problems. Indeed, if $\operatorname{diam}(S(N^{\text{non}})^{-1}(0)) = 2r(N^{\text{non}})$, then $r(N^{\text{non}}) \le r(N^{\text{a}})$. This holds, for instance, if S is a linear functional or if the codomain of the operator T, which generates F, lies in a Hilbert space. In fact, we do not know a linear problem for which $r(N^{\text{non}}) \le r(N^{\text{a}})$ does not hold.

The theorem states that the far more general structure of adaptive information cannot decrease uncertainty by more than a factor of two, as compared to the simpler structure of nonadaptive information.

The theorem has an interesting history. It was proven by Bakhvalov (1971) assuming that S is a linear functional, and by Gal and Micchelli (1980) and Traub and Woźniakowski (1980) assuming that the cardinality $n(f)$ of adaptive information is independent of f. The general case is considered by Wasilkowski (1985b), who also considers the average case setting. Generalizations for different information operators may be found in Traub, Wasilkowski, and Woźniakowski (1983).

Also, adaption does not help for linear problems for the average case setting assuming that the probability measure enjoys a certain "symmetry" property. The general case with varying $n(f)$ was proven by Wasilkowski

(1985b). The case with constant $n(f)$ was proven by Traub, Wasilkowski, and Woźniakowski (1984a, b), Wasilkowski and Woźniakowski (1984a), and Lee and Wasilkowski (1985) under various assumptions about the spaces F_1 and G.

Adaption does not help for linear problems if stochastic information is used, as established by Kadane, Wasilkowski, and Woźniakowski (1984). Here it is assumed that each evaluation is computed with some noise. The assumption is that noise is unbiased.

Also, adaption does not help for asymptotic settings as shown by Trojan (1984), and Wasilkowski and Woźniakowski (1984b).

We stress that for some nonlinear problems adaption *helps* significantly, as is briefly discussed in Section 3.8.

3.3. *Linear Algorithms*

From Section 3.2 it follows that for linear problems we need only consider nonadaptive information.

$$N(f) = [L_1(f), L_2(f), \ldots, L_n(f)].$$

We now want to find an optimal error algorithm ϕ that uses N and for which the time needed to compute $\phi(y)$ is dominated by the time needed to compute $y = N(f)$, i.e.,

$$e(\phi, N) = r(N) \qquad \text{and} \qquad \text{cost}(\phi, N(f)) \ll \text{cost}(N, f).$$

A linear algorithm is a good candidate. A *linear algorithm* ϕ^L, is an algorithm of the form

$$\phi^L(N(f)) = \sum_{i=1}^{n} L_i(f) q_i, \qquad q_i \in G.$$

Since $q_1, q_2, \ldots, q_n$ are independent of f, they can be precomputed. Then the actual computation of $\phi(N(f))$ given $N(f)$ requires n scalar multiplications by elements from the set G and $(n - 1)$ additions of elements from G. If $G = \mathbf{R}$ then n scalar multiplications and $(n - 1)$ additions of real numbers are performed.

For linear problems we are thus motivated to ask: "Does there exist a linear optimal error algorithm ϕ^L?" That is, does there exist a linear algorithm ϕ^L for which $e(\phi^L, N) = r(N)$? There is a vast literature on this subject. Linear optimal error algorithms have been found for numerous linear problems of practical importance (see Traub and Woźniakowski (1980) and the papers cited there). For instance, if S is a linear functional then there exists a linear optimal error algorithm. This is due to Smolyak (1965) for the real case (see Bakhvalov (1971) and Osipenko (1976) for the complex case). If the co-

domain of the operator T, which generates the set F, is a Hilbert space and $T(\ker N)$ is closed, then for any linear operator S, a linear optimal error algorithm exists and is given by a spline algorithm (see Micchelli and Rivlin (1977) for T, the identity operator, and Traub and Woźniakowski (1980) for general linear T).

Let us come back to the general case. Does there exist a linear optimal error algorithm for a linear problem? The answer is, in general, *no*. The first example of a linear problem for which no linear optimal error algorithm exists is due to Micchelli (1978) and can be found in Traub and Woźniakowski (1980, p. 60). Packel (1984) also presents a linear problem with no linear optimal error algorithm.

For both problems of Micchelli (1978) and Packel (1984) one can, however, observe that there exists a linear algorithm ϕ^L whose error is slightly larger than the radius of information, i.e., $e(\phi^L, N)/r(N)$ is close to one. Having this in mind, one can relax the previous question by asking: "Does there exist a constant d (which one hopes is close to one) such that

$$\inf\{e(\phi^L, N) : \phi^L \text{ is linear}\} \leq dr(N)$$

for every linear problem?"

The answer to this question is negative as is proven in a recent paper of Werschulz and Woźniakowski (1985). They exhibit a class of linear problems whose radii $r(N)$ are finite but for which the error of any linear algorithm is infinite. Furthermore, $r(N)$ can be arbitrarily small if N is appropriately chosen. One may expect that the linear problems for which this holds are artificially constructed. This is not the case, since an example of such a linear problem is the inversion of a finite Laplace transform, a problem arising in remote sensing (see Twomey, 1977).

Thus, there need not exist a linear optimal (or nearly optimal) error algorithm for every linear problem. Nevertheless, we can still achieve something, as was proven by Packel (1984). Namely, Packel (1984) proves that if the range of the solution operator is suitably extended, then linear optimal error algorithms exist for any linear problem. This indicates that the range G of the linear problem is sometimes too "small" to guarantee the existence of linear optimal error algorithms. If G is suitably extended, then the range of linear algorithms is larger and one can find a linear optimal error algorithm.

3.4. *Spline Algorithms*

Spline algorithms are optimal or almost optimal error algorithms. They are defined as follows. First, recall the definition of a spline element in a linear space (see, for instance, Anselone and Laurent, 1968; Atteia, 1965; Holmes 1972).

Let N be nonadaptive information and let $y = N(f) = [L_1(f), L_2(f), \ldots, L_n(f)]$. An element $\sigma = \sigma(y)$ is called a *spline* interpolating y iff

(i) $N(\sigma) = y$,
(ii) $\|T\sigma\| = \min\{\|Tg\| : g \in F_1, N(g) = y\}$.

Thus, $\sigma(y)$ is an element which is indistinguishable from f under the information N and which has minimal norm among all elements interpolating y. The element $\sigma(y)$ is sometimes called a *minimal norm interpolant*. The existence and properties of splines are discussed in the papers cited above.

An algorithm ϕ^s is called a *spline algorithm* iff

$$\phi^s(N(f)) = S\sigma(N(f)).$$

Spline algorithms enjoy many optimality properties (see Traub and Woźniakowski, 1980, Chap. 4). In particular, the error of a spline algorithm is always close to the radius of information,

$$e(\phi^s, N) \leq 2r(N).$$

This holds, even if S is a nonlinear operator. If the range of T lies in a Hilbert space and $T(\ker N)$ is closed, then the spline algorithm is a *linear optimal error* algorithm, i.e., $e(\phi^s, N) = r(N)$ and

$$\phi^s(N(f)) = \sum_{i=1}^{n} L_i(f) S q_i,$$

where q_i is a spline interpolating the unit vector $e_i = [0, \ldots, 1, \ldots, 0]$. Spline algorithms also play a major role in the average case setting. In this case, $F = F_1$ and the operator T which defines a spline in (ii) is given in terms of the covariance operator of a probability measure. The spline algorithm is linear and has minimal average error among all linear algorithms. If the probability measure is "orthogonally invariant" then the spline algorithm has minimal average error among all algorithms (see Wasilkowski and Woźniakowski, 1982).

Spline algorithms enjoy optimality properties for asymptotic settings as well as for different error criteria (see Trojan, 1984; Kacewicz, 1984b; Wasilkowski and Woźniakowski, 1984b; Wasilkowski, 1984).

3.5. *Optimal Information*

Recall that we can compute $L(f)$, where L is a linear functional from the given class Λ. Without loss of generality we can consider nonadaptive information since nothing can be gained by using adaptive information. Nonadaptive information $N(f) = [L_1(f), \ldots, L_n(f)]$ consists of n functionals

$L_1, L_2, \ldots, L_n$. How should we choose L_i in order to minimize the cost of computing an ϵ-approximation?

The cost of $N(f)$ is cn, where n is the cardinality of N. For fixed cardinality, we should choose L_i such that the intrinsic uncertainty due to L is minimal. The intrinsic uncertainty is, as we know, measured by the radius of information.

That is, for given integer n, we wish to find $N_n^* = [L_1^*, \ldots, L_n^*]$, $L_i^* \in \Lambda$, such that

$$r(N_n^*) = \inf\{r(N) : N = [L_1, \ldots, L_n], L_i \in \Lambda\}.$$

Such information N_n^* is called *nth optimal error information*.

We now present nth optimal error information for the class Λ of all linear functionals. Assume for simplicity that T is injective and its range lies in a Hilbert space H with the inner product $(\cdot, \cdot)$. We also assume that G is a Hilbert space. Suppose that

$$A = (ST^{-1})^* ST^{-1} : H \to H.$$

is a compact operator. Denote by $\xi_1, \xi_2, \ldots$ orthonormal eigenelements of the operator A, $A\xi_1 = \lambda_i \xi_i$, $\lambda_1 \geq \lambda_2 \geq \cdots$. Then, nth optimal error information is given by

$$N_n^*(f) = [(Tf, \xi_1), (Tf, \xi_2), \ldots, (Tf, \xi_n)].$$

The spline algorithm ϕ^s using this information is now equal to

$$\phi^s(N_n^*(f)) = \sum_{i=1}^{n} (Tf, \xi_i) ST^{-1} \xi_i.$$

The spline algorithm is linear and optimal error,

$$e(\phi^s, N_n^*) = r(N_n^*) = \lambda_{n+1}^{1/2}$$

(see Traub and Woźniakowski, 1980, Chaps. 2–4).

The ϵ-complexity can be determined from nth optimal error information as we shall see in the next section. The problem of nth optimal error information is related to Gelfand n-widths of the set $S(F)$. If there exists a linear optimal error algorithm, it is also related to Kolmogorov n-widths of the set $S(F)$ (see Traub and Woźniakowski, 1980, p. 41). The reader is referred to a recent book of Pinkus (1985), where the study of various n-widths may be found.

3.6. *Complexity for Linear Problems*

In order to find an ϵ-approximation we must use information N such that $r(N) \leq \epsilon$. Since adaption does not help for linear problems, N can be chosen

to be nonadaptive. That is, $N = [L_1, L_2, \ldots, L_n]$, $L_i \in \Lambda$. Its cardinality has to be at least equal to the ϵ-cardinality as defined in Section 2.6,

$$m(\epsilon) = \min\{\text{card}(N) : r(N) \leq \epsilon\}.$$

This means that we should use nth optimal error information N_n^* with $n = m(\epsilon)$. In Hilbert spaces with injective T and compact A, Section 3.5 yields

$$m(\epsilon) = \min\{n : \lambda_{n+1}^{1/2} \leq \epsilon\}.$$

Since in this case the spline algorithm has minimal error and is linear, we conclude that information N_n^* and the spline algorithm ϕ^s are almost *optimal*,

$$\text{comp}(\epsilon) \sim \text{cost}(\phi^s, N_n^*) \sim cm(\epsilon).$$

We can choose operators S and T such that λ_i goes to zero arbitrarily slowly. Then, the ϵ-cardinality $m(\epsilon)$ can go to infinity arbitrarily fast as ϵ goes to zero. Thus, there exist linear problems with arbitrarily large complexity. Furthermore, there are no "gaps" in the complexity functions (see Traub and Woźniakowski, 1980, Chap. 5). That is, for any increasing function g one can find a linear problem for which

$$\text{comp}(e) \sim cg(1/\epsilon), \qquad \epsilon \to 0.$$

This may be contrasted with the theory of recursively computable functions in which complexity gaps are known to occur (see Borodin, 1972).

3.7. *Linear Problems of Special Interest*

Optimal error algorithms for certain linear problems have been the subject of intensive study. Probably the most popular one is the integration problem. There are about 100 papers cited in Traub and Woźniakowski (1980, Chap. 6), where the integration problem is analyzed for many different classes of functions. In most cases, the analysis is done for the scalar case.

The problem of approximation, $S(f) = f$, has been also widely studied. In the book edited by Babenko (1979), the multivariate approximation problem is discussed, i.e., f is a function of k variables. The dependence of the ϵ-complexity on the smoothness of the functions and the number of variables is obtained.

As a sample of recent work on linear problems we mention three examples. The first is the approximate solution of partial differential or integral equations $Pu = f$. Here the linear partial differential or integral operator P and its domain are fixed and the right-hand side function f belongs to some balanced and convex set F. Information about f is given by its function values or by inner products. Then $S(f) = P^{-1}(f)$ is a linear problem. In a number of

recent papers Werschulz (1983b, 1985a) analyzes this problem and shows when a finite element method is an optimal error algorithm. The ϵ-complexity is also obtained. The reader is referred to the survey paper, Werschulz (1985b).

Kowalski (1985) considers a signal processing problem. He finds the radius of information and the ϵ-complexity for the approximation of band and energy limited signals.

Lee (1985) discusses a number of image understanding problems which arise in computer vision. He analyzes them from the viewpoint of information-based complexity.

3.8. *Nonlinear Problems*

We end this section by indicating work on a number of nonlinear problems. Not surprisingly, the results here are less general than for linear problems.

One of the first papers in information-based complexity analyzed a nonlinear problem. This is the paper of Kiefer (1953), which contains the results of his Master Thesis from 1948. He showed that if function evaluations are used, Fibonacci search is optimal in searching for the maximum of a unimodal function.

It is known that for some nonlinear problems adaption *does* help significantly. An example of such a problem is zero finding for scalar real continuous functions which change sign at the endpoints of the domain (see Sikorski, 1982).

As a sample of recent work on nonlinear problems we mention three examples. The first is the approximate solution of scalar or multivariate nonlinear equations. A survey of recent results may be found in Sikorski (1985).

The second example is the approximate solution of ordinary differential equations $z'(t) = f(t, z(t))$ for $t \in (0, 1]$ and known $z(0)$. Here partial information is assumed about f. The solution operator is defined as the solution of the ODE problem, $S(f) = z$. This problem is nonlinear since z depends nonlinearly on f. Kacewicz (1982,1983,1984a) analyzes this problem thoroughly, admitting even nonlinear continuous operations on f. He shows how the smoothness of f and the type of permissible information determine the ϵ-complexity.

The third example is the eigenvalue problem where the approximate eigenpairs are sought for matrices of large size. Kuczyński (1985) analyzes this problem for Krylov information, that is, for information $N_k(A, b) = [b, Ab, \ldots, A^k b]$ for some nonzero vector b. This information is widely used in practice since it can be efficiently computed for sparse matrices. Kuczyński proves that the generalized minimal residual algorithm almost minimizes the number of matrix–vector multiplications needed to find an ϵ-approximation. He also shows that the widely used Lanczos algorithm is far from optimal. Chou (1985) uses Nemirovsky and Yudin (1983) to show quasi

optimality of Krylov information and uses Kuczyński (1985) to derive the ϵ-complexity for the class of symmetric matrices with $\|A\| \leq 1$. The ϵ-complexity is proportional to $1/\epsilon$ for matrices of size larger than $1/\epsilon$.

We close this section with a few comments on an important book of Nemirovsky and Yudin (1983), mentioned several times above. The book is devoted to the analysis of the nonlinear constrained optimization problem

$$\min\{f_0(x) : x \in G, f_j(x) \leq 0, j = 1, 2, \ldots, m\}.$$

Here G is a subset of a real Banach space, and f_j, for $j = 0, 1, \ldots, m$, are continuous scalar functions belonging to some class F. The authors find sharp estimates on the minimal number of function and first derivative evaluations in order to determine the minimum to within ϵ. They analyze the classes of convex and strongly convex functions as well as the classes of nonconvex smooth functions. This analysis is done for a convex and/or compact set G. They find bounds on the ϵ-cardinality numbers which quantify the relative value of convexity as opposed to smoothness. The authors do not study combinatory cost and therefore only lower bounds on the ϵ-complexity are obtained. An interesting open problem is to find the ϵ-complexity of these nonlinear optimization problems.

4. Average Case Setting

So far we discussed information-based complexity in the worst case setting. That is, the error and the cost were defined by worst performance. A more realistic approach might be to define the error and the cost on the average. This can be done as follows.

Recall we want to compute $x_\epsilon = x_\epsilon(f)$, which approximates $S(f)$ for f from the set F. Assume that F is equipped with a *probability measure* μ. How do we choose a "good" measure μ? Assume first that F is finite, $F = \{f_1, f_2, \ldots, f_k\}$. Let $p(f)$ denote the probability of occurrence of an element f, $p(f) \geq 0$ and $\Sigma_{f \in F} p(f) = 1$. Then for $A \subseteq F$, $\mu(A)$ is given by $\Sigma_{f \in A} p(f)$.

Assume now that F is a subset of the k-dimensional euclidean space, $F \subseteq \mathbf{R}^k$. Then a natural choice is a weighted Lebesgue measure, $\mu(A) = \int_A p(f)\, df$.

However, if F lies in an infinite-dimensional space, the choice of measure μ is not obvious, since there is no Lebesgue-type measure in an infinite-dimensional space. We stress that many information-based complexity problems are defined on infinite-dimensional spaces. For such problems, infinite-dimensional measure must be used. We believe that measures such as Wiener, or more generally Guassian, measures, which are commonly used in many applied fields, may serve as good candidates for the average case setting.

Having agreed which measure μ should be chosen, we now proceed as follows. Let N be partial information and let ϕ be an algorithm using N. The *average error* of ϕ is defined as

$$e^{\text{avg}}(\phi, N) = \int_F \|S(f) - \phi(N(f))\| \mu(df).$$

The *average cost* of ϕ is given by

$$\text{cost}^{\text{avg}}(\phi, N) = \int_F \{\text{cost}(N, f) + \text{cost}(\phi, N(f))\} \mu(df).$$

Then the *average ϵ-complexity* is defined as

$$\text{comp}^{\text{avg}}(\epsilon) = \inf\{\text{cost}^{\text{avg}}(\phi, N) : \phi, N \text{ such that } e^{\text{avg}}(\phi, N) \leq \epsilon\}.$$

That is, the average ϵ-complexity is the minimal average cost of finding an approximation whose average error does not exceed ϵ. A survey of recent results for the average case setting may be found in Wasilkowski (1985a).

4.1. *Average Case Analysis of Continuous Binary Search*

We end this section by analyzing the continuous binary search problem on the average. As far as we know the analysis is new.

EXAMPLE (Continuous Binary Search—*continued*). We discussed this problem in Example 1.1. The (worst case) ϵ-complexity is

$$\text{comp}(\epsilon) = c\lceil \log_2(1/\epsilon) - 1 \rceil.$$

We show that the average ϵ-complexity is essentially the same. First, we choose our measure μ to be Lebesgue measure on (0,1). That is, for a Borel set $A \subseteq (0, 1)$, $\mu(A) = \int_A dx$. Recall that we want to find $x \in (0,1)$ by asking questions. Let

$$N(x) = [Q(x; T_1), Q(x; T_2), \ldots, Q(x; T_{n(x)})] \tag{1}$$

denote the questions we ask about the number x. Here T_i is a Borel subset of $(0, 1)$ and the choice of T_i as well as the number $n(x)$ of questions may depend on the previous answers. Thus, N is *adaptive* information.

Knowing $N(x)$, we find an approximation x_ϵ by some algorithm ϕ, ϕ: $N(0, 1) \to (0, 1)$. That is, $x_\epsilon = \phi(N(x))$. The average error is now given by

$$e^{\text{avg}}(\phi, N) = \int_0^1 |x - \phi(N(x))| \, dx,$$

and the average cost

$$\text{cost}^{\text{avg}}(\phi, N) \geq \int_0^1 \{cn(x) + \text{cost}(\phi, N(x))\}\, dx,$$

where c is the cost of one question. To find the average ϵ-complexity we have to solve the minimization problem

$$\text{comp}^{\text{avg}}(\epsilon) = \min\{\text{cost}^{\text{avg}}(\phi, N) : \phi, N \text{ such that } e^{\text{avg}}(\phi, N) \leq \epsilon\}.$$

Assume first that $\epsilon \geq \frac{1}{4}$. Set $x_\epsilon = \frac{1}{2}$. The average error is now equal to

$$\int_0^1 |x - \tfrac{1}{2}|\, dx = \tfrac{1}{4} \leq \epsilon.$$

Thus, we can solve the problem with cost = 0, i.e., $\text{comp}^{\text{avg}}(\epsilon) = 0$ for $\epsilon \geq \frac{1}{4}$.

Let $\epsilon < \frac{1}{4}$. Consider nonadaptive information N_k with $k = \lceil \log_2(1/\epsilon) - 2 \rceil$ as in Example 1.1. That is, the ith question is given by: "Is the ith bit of x zero?" Then

$$\phi_k(N_k(x)) = \sum_{i=1}^{k} b_i 2^{-1} + 2^{-(k+1)}, \tag{2}$$

where $b_i = 0$ if the ith answer is yes, and $b_i = 1$ otherwise. It can be checked directly that

$$e^{\text{avg}}(\phi_k, N_k) = \int_0^1 |x - \phi_k(N_k(x))|\, dx = 2^{-(k+2)} \leq \epsilon.$$

Thus, the average ϵ-complexity is no higher than the average cost of ϕ_k and N_k,

$$\text{comp}^{\text{avg}}(\epsilon) \leq c\lceil \log_2(1/\epsilon) - 2 \rceil. \tag{3}$$

We now show that the bound (3) cannot be substantially improved. First we need the estimate

$$\inf_a \int_A |x - a|\, dx \geq \mu^2(A)/4 \tag{4}$$

for any Borel set A. (Here $\mu(A)$ denotes Lebesgue measure of A.) Indeed, (4) can be directly verified if A is an interval. Then using simple geometrical

arguments it can be shown that (4) holds for A being a countable union of disjoint intervals. Therefore (4) is true for any Borel set.

Now take adaptive information (1) with fixed $n(x) \equiv n$. Let ϕ be an algorithm using N. Observe the $N(x)$ takes p different values, where $p \leq 2^n$. Thus, the interval (0,1) can be partitioned into p disjoint subsets A_i, $i = 1, 2, \ldots, p$, such that $\phi(N(x))$ is constant for $x \in A_i$. Denote its value on A_i by c_i, $c_i = \phi(N(x))$ for $x \in A_i$. Then (4) yields

$$e^{\text{avg}}(\phi, N) = \sum_{i=1}^{p} \int_{A_i} |x - c_i| \, dx \geq \sum_{i=1}^{p} \mu^2(A_i)/4 = \left(\sum_{i=1}^{p} \mu(A_i) \right)^2 /(4p)$$
$$= 1/(4p) \geq 2^{-(n+2)}. \tag{5}$$

Thus, the average error of ϕ has to be at least $2^{-(n+2)}$. This bound is sharp. For instance, for N_k and ϕ_k defined by (2) we obtain equality with $n = k$.

Consider now the general case, i.e., adaptive information N with varying $n(x)$. Let B_i denote the subset of (0, 1) for which $n(x) = i$. Then

$$e^{\text{avg}}(\phi, N) = \sum_{i=1}^{\infty} \int_{B_i} |x - \phi(N(x))| \, dx.$$

For $x \in B_i$, $\phi(Nx))$ takes at most p_i, $p_i \leq 2^i$, different values. Let B_i be the union of disjoint $A_{i,j}$, $j = 1, 2, \ldots, p_i$. Then $\phi(N(x)) \equiv c_{i,j}$ for $x \in A_{i,j}$. From (4) we have

$$e^{\text{avg}}(\phi, N) = \sum_{i=1}^{\infty} \sum_{j=1}^{p_i} \int_{B_i \cap A_{ij}} |x - c_{i,j}| \, dx \geq \sum_{i=1}^{\infty} \sum_{j=1}^{p_i} \mu^2(A_{i,j})/4$$
$$\geq \sum_{i=1}^{\infty} (4p_i)^{-1} \left(\sum_{j=1}^{p_i} \mu(A_{i,j}) \right)^2 \geq \sum_{i=1}^{\infty} 2^{-i} \mu^2(B_i)/4.$$

The average cost is given by

$$\text{cost}^{\text{avg}}(\phi, \text{N}) \geq \text{c} \int_0^1 n(x) \, dx = c \sum_{i=1}^{\infty} i\mu(B_i) \geq cf(\epsilon),$$

where

$$f(\epsilon) = \min \left\{ \sum_{i=1}^{\infty} i\mu(B_i) : \sum_{i=1}^{\infty} 2^{-i} \mu^2(B_i)/4 \leq \epsilon, \sum_{i=1}^{\infty} \mu(B_i) = 1 \right\}.$$

For a positive z define

$$g(z) = \min \left\{ \sum_{i=1}^{\infty} 2^{-i} \mu^2(B_i)/4 : \sum_{i=1}^{\infty} i\mu(B_i) = z, \sum_{i=1}^{\infty} \mu(B_i) = 1 \right\}.$$

The value of $g(z)$ can be found using a standard technique, $g(z) \sim 2^{-(z+2)}$ for large z. Since $f(\epsilon) = g^{-1}(\epsilon)$, we have

$$f(\epsilon) = (\log_2(1/\epsilon) - 2)(1 + o(1)), \qquad \text{as } \epsilon \to 0.$$

This and (3) prove that

$$\text{comp}^{\text{avg}}(\epsilon) = c[\log_2(1/\epsilon) - 2](1 + o(1)), \qquad \text{as } \epsilon \to 0.$$

5. Final Comments

5.1. *Settings of Information-Based Complexity*

We have briefly discussed two settings in information-based complexity: the worst case and average case settings. There are a number of different settings which are also of importance. One such setting is a *probabilistic* setting in which one seeks an ϵ-approximation for a subset of F which has large measure. The reader is referred to Wasilkowski (1984), where the analysis of such a setting can be found.

A different setting is provided by an *asymptotic* setting in which one wants to approximate $S(f)$ by a sequence of approximations with best possible speed of convergence. The asymptotic setting was analyzed by Trojan (1984) for linear problems and by Kacewicz (1984b) for nonlinear problems. They did not assume that the space of elements f is equipped with a measure and showed a surprising relation between this asymptotic setting and the worst case one. The asymptotic setting was also analyzed by Wasilkowski and Woźniakowski (1984b), assuming that the space of elements f is equipped with a Gaussian measure. They showed a relation between the asymptotic and the average case settings.

So far, the error between $S(f)$ and an approximation x_ϵ was determined by $\|S(f) - x_\epsilon\|$. Sometimes, a different error criterion is appropriate. For instance, one may consider the relative error $\|S(f) - x_\epsilon\|/\|S(f)\|$, or a combination of absolute and relative errors, $\|S(f) - x_\epsilon\|/(\|S(f)\| + \nu)$ for some positive ν. In general, one can consider some error functional $E\colon F \times G \to \mathbf{R}_+$. Then the error is determined by $E(f, x_\epsilon)$. Sometimes the error functional E depends only on the difference $S(f) - x_\epsilon$, i.e., $E(f, x_\epsilon) = H(S(f) - x_\epsilon)$ for some $H\colon G \to \mathbf{R}_+$. The reader may find the analysis of different error criteria in Traub, Wasilkowski, and Woźniakowski (1983, 1984a), Wasilkowski (1984), and Lee and Wasilkowski (1985).

5.2. *Models of Computation*

We have presented several examples where the real number model was assumed as the model of computation. That is, real numbers are used, and information and combinatory operations are performed with infinite precision and unit cost.

For scientific computation, fixed precision floating point arithmetic is almost universally used. In the fixed precision model, we have an additional source of error due to round-off, and numerical stability becomes an important issue.

We use the real number model rather than the fixed precision floating point to avoid being distracted by round-off issues. The numerical stability of optimal error algorithms should be studied. Are there the optimality–stability trade-offs? If an optimal error algorithm is numerically stable, then complexity results for the fixed precision model are essentially the same as those for the real number model. A more detailed discussion on the real number and fixed precision models may be found in Traub and Woźniakowski (1982).

We also assumed *sequential* computations. That is, one operation is allowed at a time and the total cost is the sum of the operation costs. *Parallel* or *distributed* computations may be also studied. Then one can perform a number of operations simultaneously and the total cost depends on the number of parallel steps and not on the total number of operations.

We assumed that the cost of information operations is fixed and does not depend on a specific operation or a problem element. For instance, if function evaluations are permitted then the cost of computing $f(x)$ is assumed to be fixed independently of a function f and a point x. This means, that we are really charged for the use of a subroutine call independent of the body of the subroutine and its input. In actual computation the cost of computing $f(x)$ may depend on f, x as well as on the required precision.

We also assumed that the cost of combinatory operations is fixed. For example, the cost of addition or multiplication of two real numbers is assumed to be the same. This assumption has been made for simplicity.

The study of information-based complexity with different models of computation is a rich area of future research. Models which should be studied include those with one or more of the following characteristics:

—computation is parallel or distributed;

—cost of information operations depends on the specific operation, problem element, or precision;

—cost of combinatory operations depends on the operation or precision.

5.3. *Future Directions*

A number of directions for future research dealing with models of computation were presented in the previous section. Additional ideas are given here. As we already indicated in the Introduction, one of the major future directions of information-based complexity should be the study of contaminated information. That is, one assumes that one knows the contaminated information $N(f) + e$ rather than $N(f)$. Sometimes the bound on the noise e is known, as in Micchelli and Rivlin (1977) and Traub, Wasilkowski, and Woźniakowski (1983). Sometimes the noise e is assumed to be a random variable with known distribution, as in Kadane, Wasilkowski, and Woźnia-

kowski (1984). The contaminated information should be studied for a number of different settings. "Mixed" settings seem to be of practical importance. For instance, one can use a worst case setting with respect to problem elements f and an average case setting with respect to the noise e.

Another future direction is the study of problems whose formulation is only partially known. So far we assumed that the solution operator S and the class F of problem elements were exactly given. For the average case we also assumed that the probability measure μ is exactly known. Sometimes, we may have only *partial* information about S, F, and μ.

For instance, one may know only some properties of μ such as its mean element and its covariance operator. This problem was analyzed by Kadane and Wasilkowski (1984) and Wasilkowski (1985a).

As the next example consider partial information on F. We call this the "fat" F problem. To explain the fat F problem, assume that f is a scalar function. We know that f is smooth but we do not know exactly how many times f is differentiable. So, $f \in F_r$ for a number of values of r, where F_r denotes a class of r times differentiable functions. We would like to find an algorithm ϕ that works well for all F_r under consideration. Let $e(\phi, N, F_r)$ denote the error of algorithm ϕ using information N for problem elements from the class F_r. The algorithm ϕ is good for the class F_r if its error is close to the radius of information $r(N, F_r)$. So, we seek an algorithm ϕ for which $e(\phi, N, F_r)/r(N, F_r)$ is close to one for all classes F_r under consideration. Whether such an algorithm exists depends on the problem. For example, in the integration problem 1.3, the midpoint rule is an optimal error algorithm for all F_r with $r = 1, 2, \ldots$. What characterizes problems for which a single algorithm is optimal for a number of classes? This and similar questions will be focus of future research.

ACKNOWLEDGMENTS

I wish to express my thanks to J. Kuczyński, K. Sikorski, G. W. Wasilkowski, and A. Werschulz for valuable comments and suggestions concerning this paper.

REFERENCES

ANSELONE, P. M., AND LAURENT, P. J. (1968), A general method for the construction of interpolating or smoothing spline functions, *Numer. Math.* **12,** 66–82.

ATTEIA, M. (1965), Fonctions-spline généralisées, *C. R. Acad. Sci. Paris* **216,** 2149–2152.

BABENKO, K. I. (1979), "Theoretical Background and Constructing of Computational Algorithms for Mathematical-Physical Problems" (K. I. Babenko, Ed.), Nauka, Moscow.

BAKHVALOV, N. S. (1971), On the optimality of linear methods for operator approximation in convex classes of functions, *Zh. Vychisl. Mat. i Mat. Fiz.* **11,** 1014–1018 [Russian]; *Comput. Math. Math. Phys.* **11** 244–249 [English transl.]

BORODIN, A. (1972), Complexity classes of recursive functions and the existence of complexity gaps, *J. Assoc. Comput. Mach.* **19,** 158–174, 576.

CHOU, A. (1985), On the optimality of Krylov information, in progress.

GAL, S., AND MICCHELLI, C. A. (1980), Optimal sequential and non-sequential procedures for evaluating a functional, *Appl. Anal.* **10,** 105–120.

GOLOMB, M., AND WEINBERGER, H. F. (1959), Optimal approximation and error bounds, *in* "On Numerical Approximation" (R. E. Langer, Ed.), pp. 117–190, Univ. of Wisconsin Press, Madison.

HOLMES, R. (1972), R-Splines in Banach spaces, I. Interpolation of linear manifolds, *J. Math. Anal. Appl.* **40,** 574–593.

KACEWICZ, B. Z. (1982), On the optimal error of algorithms for solving a scalar autonomous ODE, *BIT* **22,** 503–518.

KACEWICZ, B. Z. (1983), Optimality of Euler-integral information for solving a scalar autonomous ODE, *BIT* **23,** 217–230.

KACEWICZ, B. Z. (1984a), How to increase the order to get minimal-error algorithms for systems of ODE, *Numer. Math.* **45,** 93–104.

KACEWICZ, B. Z. (1984b), "Asymptotic Error of Algorithms for Solving Nonlinear Problems," IRMA-Istituto per le Ricerche Di Matematica Aplicata Del CNR, Bari.

KADANE, J. B., AND WASILKOWSKI, G. W. (1984), Average case ϵ-complexity: A Bayesian view *in* "Proceedings, II Valencia International Meeting on Bayesian Statistics."

KADANE, J. B., WASILKOWSKI, G. W., AND WOŹNIAKOWSKI, H. (1984), "Can Adaption Help on the Average for Stochastic Information?" Report, Columbia University.

KIEFER, J. (1953), Sequential minimax search for a maximum, *Proc. Amer. Math. Soc.* **4,** 502–505.

KOWALSKI, M. A. (1985), Optimal recovery of band and energy limited signals, in preparation.

KUCZYŃSKI, J. (1985), A generalized minimal residual algorithm for finding an eigenpair of a large matrix, *J. Complexity,* to appear.

LEE, D. (1985), Optimal algorithms for image understanding: Current status and future plans, *J. Complexity,* **1,** 138–146.

LEE, D., AND WASILKOWSKI, G. W. (1985), Approximation of linear functionals on a Banach space with a Gaussian measure, *J. Complexity,* in press.

MICCHELLI, C. A., AND RIVLIN T. J. (1977), A survey of optimal recovery, *in* "Optimal Estimation in Approximation Theory" (C. A. Micchelli and T. J. Rivlin, Eds.), pp. 1–54, Plenum, New York.

MOTORNYJ, V. P. (1973), On the best quadrature formula of the form $\Sigma_{k=1}^{n} p_k f(x_k)$ for some classes of periodic differentiable functions, *Dokl. Akad. Nauk SSSR* **211,** 1060–1062 [Russian]; *Soviet. Math. Dokl.* **14,** 1180–1183 [English transl.]

NEMIROVSKY, A. S., AND YUDIN, D. B. (1983), "Problem Complexity and Method Efficiency in Optimization," Wiley–Interscience, New York.

NIKOLSKIJ, S. M. (1950), On the problem of approximation estimate by quadrature formulae, *Uspekhi Mat. Nauk* **5,** 165–177.

OSIPENKO, K. YU. (1976), Best approximation of analytic functions from information about their values at a finite number of points, *Mat. Zametki* **19,** 29–40 [Russian]; *Math. Notes* **19,** 17–23 [English transl.]

PACKEL, E. W. (1984), Linear problems (with extended range) have linear optimal algorithms, *Aequationes Math.*, to appear.

PAPADIMITRIOU, C. H., AND TSITSIKLIS, J. (1984), Intractable problems in control theory, submitted for publication.

PINKUS, A. (1985), "n-Widths in Approximation Theory," Springer-Verlag, Berlin.

RIVEST, R. L., MEYER, A. R., KLEITMAN, D. J., WINKLMANN, K., AND SPENCER, J. (1980), Coping with errors in binary search procedures, *J. Comput. System Sci.* **20,** 396–404.

SARD, A. (1949), Best approximate integration formulas; best approximation formulas, *Amer. J. Math.* **71,** 80–91.

SIKORSKI, K. (1982), Bisection is optimal, *Numer. Math.* **40,** 111–117.

SIKORSKI, K. (1985), Optimal solution of nonlinear equations, *J. Complexity,* in press.

SMOLYAK, S. A. (1965), "On Optimal Restoration of Functions and Functionals of Them," Candidate Dissertation, Moscow State University. [Russian]

TRAUB, J. F. (1985a), Information, complexity, and the sciences, University Lecture, Columbia University, 1985.

TRAUB, J. F. (1985b), Complexity of approximately solved problems, *J. Complexity* **1,** 3–10.

TRAUB, J. F., WASILKOWSKI, G. W., AND WOŹNIAKOWSKI, H. (1983), "Information, Uncertainty, Complexity," Addison–Wesley, Reading, Mass.

TRAUB, J. F., WASILKOWSKI, G. W., AND WOŹNIAKOWSKI, H. (1984a), Average case optimality for linear problems, *J. TCS* **29,** 1–25.

TRAUB, J. F., WASILKOWSKI, G. W., AND WOŹNIAKOWSKI, H. (1984b), When is nonadaptive information as powerful as adaptive information? *in* "Proceedings, 23rd IEEE Conference on Decision and Control," pp. 1536–1540.

TRAUB, J. F., AND WOŹNIAKOWSKI, H. (1980), "A General Theory of Optimal Algorithms," Academic Press, New York.

TRAUB, J. F., AND WOŹNIAKOWSKI, H. (1982), Complexity of linear programming, *Oper. Res. Lett.* **1,** 59–62.

TRAUB, J. F., AND WOŹNIAKOWSKI, H. (1984a), Information and computation, *in* "Advances in Computers, Vol. 23" (M. C. Yovits, Ed.), pp. 35–92, Academic Press, New York.

TRAUB, J. F., AND WOŹNIAKOWSKI, H. (1984b), On the optimal solution of large linear systems, *J. Assoc. Comput. Mach.* **31,** 545–559.

TROJAN, G. M. (1984), Asymptotic setting for linear problems, manuscript.

TWOMEY, S. (1977), "Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurement," Developments in Geomathematics, Vol. 3, Elsevier, Amsterdam.

WASILKOWSKI, G. W. (1983), "Local Average Error," *Report,* Columbia University.

WASILKOWSKI, G. W. (1984), Optimal algorithms for linear problems with Gaussian measures, *Rocky Mountain J. Math.,* to appear.

WASILKOWSKI, G. W. (1985a), Average case optimality, *J. Complexity* **1,** 107–117.

WASILKOWSKI, G. W. (1985b), Information of varying cardinality, in progress.

WASILKOWSKI, G. W., AND WOŹNIAKOWSKI, H. (1984a), Can adaption help on the average?" *Numer. Math.* **44,** 169–190.

WASILKOWSKI, G. W., AND WOŹNIAKOWSKI, H. (1984b), "On Optimal Algorithms in an Asymptotic Model with Gaussian Measure," Report, Columbia University.

WERSCHULZ, A. G. (1983a), Measuring uncertainty without a norm, *Aequationes Math.* **26,** 74–82.

WERSCHULZ, A. G. (1983b), Complexity of indefinite elliptic problems, *Math. Comp.,* to appear.

WERSCHULZ, A. G. (1985a), What is the complexity of elliptic systems? **45,** 69–84. *J. Approx. Theory.,* to appear.

WERSCHULZ, A. G. (1985b), Complexity of differential and integral equations, *J. Complexity,* in press.

WERSCHULZ, A. G., AND WOŹNIAKOWSKI, H. (1985), "Are Linear Algorithms Always Good for Linear Problems?" Report, Columbia University.